```
STMFV <- function (sv, Rn=6378137, Rm=6335439, .aaframe='a') {
  Cradeg <- pi / 180  ## conversion to radians from degrees
  OmegaE <- 2*pi/86400  ## seems to match INS better than exact value
  stmf <- vector('numeric', length=15)

  ## get the transformation matrix a-frame to l-frame
  rlm <- XformLA (data.frame(PITCH=sv[7]/Cradeg, ROLL=sv[8]/Cradeg,
                             THDG=sv[9]/Cradeg))

  ## This is the inertial correction to rotation rates; cf. Noureldin p. 179-180
  omega <- as.vector (c(-sv[5] / Rm,
                        OmegaE * cos(sv[1]) + sv[4] / (Rn),
                        OmegaE * sin(sv[1]) + sv[4] * tan(sv[1]) / Rn),
                      mode='numeric')
  ## signs account for b-frame to a-frame: reverse sign of 3, swap 1 and 2
  Oill <- matrix (c(0, omega[3], omega[1],
                    -omega[3], 0, -omega[2],
                    -omega[1], omega[2], 0), ncol=3)
  Oilb <- Oill %*% rlm

  ## find the derivative of the transformation matrix:
  ## SRM is skew-symmetric representation of measured rotation rates
  SRR <- c(0, -sv[12], -sv[10],
           sv[12], 0, sv[11],
           sv[10], -sv[11], 0)
  SRM <- aperm( array (SRR, dim=c(3,3)))
  dRLA <- rlm %*% SRM - Oilb

  ## position derivatives
  DR <- c(sv[5] / Rm, sv[4] / (Rn * cos (sv[1])), sv[6])

  ## accelerations = velocity derivatives
  Grav <- as.numeric (Gravity (sv[1] / Cradeg, sv[3]))
  AA <- as.vector (c (sv[14], sv[13], sv[15] + Grav), mode='numeric') # a-frame
  AL <- as.vector (rlm %*% as.matrix (AA), mode='numeric')   # l-frame
  ## correct for angular effects
  VL <- c(sv[4], sv[5], sv[6])
  svdf <- data.frame(LAT=sv[1] / Cradeg, GGALT=sv[3])
  AL <- as.vector (AL - RotationCorrection (svdf, VL), mode='numeric')
  AL[3] <- -AL[3] - Grav

  ## attitude-angle derivatives: see workflow document
  if (abs(rlm[1,2]) < 0.001) {  ## avoiding rounding errors; see workflow
    RRT <- c(-dRLA[3,1]/sqrt(1-rlm[3,1]^2),
             1/(1+(rlm[3,2]/rlm[3,3])^2) * (rlm[3,2]/rlm[3,3]) *
               (-dRLA[3,2]/rlm[3,2] + dRLA[3,3]/rlm[3,3]),
             (rlm[1,2]*dRLA[1,1]/rlm[1,1]^2 - dRLA[1,2]/rlm[1,1]))
  } else {
    RRT <- c(-dRLA[3,1]/sqrt(1-rlm[3,1]^2),
             1/(1+(rlm[3,2]/rlm[3,3])^2) * (rlm[3,2]/rlm[3,3]) *
               (-dRLA[3,2]/rlm[3,2] + dRLA[3,3]/rlm[3,3]),
             1/(1+(rlm[1,1]/rlm[1,2])^2) *
               (dRLA[1,1]/rlm[1,2] - dRLA[1,2]*rlm[1,1]/rlm[1,2]^2))
  }

  stmf <- c(DR, AL, RRT, rep(0, 6))
    return (as.vector (stmf, mode='numeric'))
}
```